

# Constructions of hamiltonian graphs with bounded degree and diameter $O(\log n)$ <sup>☆</sup>

Aleksandar Ilić<sup>a,\*</sup>, Dragan Stevanović<sup>b,c</sup>

<sup>a</sup> Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia

<sup>b</sup> University of Primorska—FAMNIT, Glagoljaška 8, 6000 Koper, Slovenia

<sup>c</sup> Mathematical Institute, Serbian Academy of Science and Arts, Knez Mihajlova 36, 11000 Belgrade, Serbia

## ARTICLE INFO

### Article history:

Received 2 June 2009

Accepted 8 June 2009

### Keywords:

Hamiltonian cycle

Token ring

Diameter

Binary tree

Graph algorithm

## ABSTRACT

Token ring topology has been frequently used in the design of distributed loop computer networks and one measure of its performance is the diameter. We propose an algorithm for constructing hamiltonian graphs with  $n$  vertices, maximum degree  $\Delta$  and diameter  $O(\log n)$ , where  $n$  is an arbitrary number. The number of edges is asymptotically bounded by  $(2 - \frac{1}{\Delta-1} - \frac{(\Delta-2)^2}{(\Delta-1)^3})n$ . In particular, we construct a family of hamiltonian graphs with diameter at most  $2\lceil \log_2 n \rceil$ , maximum degree 3 and at most  $1 + 11n/8$  edges.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

An undirected graph  $G = (V, E)$  can be used as a mathematical model for computer networks, where  $V$  is the set of vertices and  $E$  is the set of edges. The number of edges adjacent to a vertex  $v$  is called the degree of the vertex  $v$ . A graph is regular if all vertices have equal degrees. The distance  $d(v, u)$  between two vertices  $v$  and  $u$  is the number of edges on a shortest path between  $v$  and  $u$ . The diameter  $D$  of the graph is the maximum distance between any pair of vertices:  $D = \max_{v, u \in V} d(v, u)$ . A cycle is a sequence of three or more vertices such that two consecutive vertices are adjacent and with no repeated vertices other than the start and end vertices. A *hamiltonian cycle* is a cycle that visits each vertex of a graph exactly once. A graph  $G$  is 1-hamiltonian if, after removing an arbitrary vertex or an edge, it still remains hamiltonian. A 1-hamiltonian graph  $G$  is optimal if it contains the least number of edges among all 1-hamiltonian graphs with the same number of vertices as  $G$ .

Networks with at least one ring structure (hamiltonian cycle) are called loop networks. Distributed loop networks are extensions of ring networks and are widely used in the design and implementation of local area networks and parallel processing architectures. There are many mutually conflicting requirements when designing the topology of a computer network. For example, no pair of processors should be too far apart in order to support efficient parallel computation demands. The hamiltonian property is one of the major requirements. The *token passing* is a channel access method where data is transmitted sequentially from one ring station to the next with a control token circulating around the ring controlling access.

<sup>☆</sup> Supported by the Research Program P1-0285 of Slovenian Agency for Research and the Grant 144007 of Serbian Ministry of Science and Technological Development.

\* Corresponding author.

E-mail addresses: [aleksandari@gmail.com](mailto:aleksandari@gmail.com) (A. Ilić), [dragance106@yahoo.com](mailto:dragance106@yahoo.com) (D. Stevanović).

**Table 1**Families of 1-hamiltonian graphs and diameter  $O(\log n)$ .

| Reference       | Name           | $n$               | $D$         | Comment               |
|-----------------|----------------|-------------------|-------------|-----------------------|
| Wang et al. [4] | Eye graph      | $6 \cdot 2^s - 6$ | $O(\log n)$ | 3-connected           |
| Hung et al. [5] | Christmas tree | $3 \cdot 2^s - 2$ | $2s$        | Hamiltonian connected |
| Kao et al. [6]  | Brother tree   | $6 \cdot 2^s - 4$ | $2s + 1$    | Bipartite             |

An open problem considered in a survey [1] on distributed loop networks is following: Find hamiltonian networks,  $\Delta$ -regular on  $n$  vertices with a diameter of order  $O(\log n)$ . This problem is related to the famous  $(n, \Delta, D)$  problem in which we want to construct a graph of  $n$  vertices with maximum degree  $\Delta$  such that the diameter  $D$  is minimized, but hamiltonicity is not an issue [12]. The lower bound on the diameter  $D$  is called the *Moore bound*,

$$D \geq \log_{\Delta-1} n - \frac{2}{\Delta}.$$

Harary and Hayes [2] presented a family of optimal 1-hamiltonian planar graphs on  $n$  vertices. Wang, Hung and Hsu [3] presented another family of optimal 1-hamiltonian graphs, each of which is planar, hamiltonian, cubic, and of diameter  $O(\sqrt{n})$ . In the literature three other families of cubic, planar and optimal 1-hamiltonian graphs with diameter  $O(\log n)$  are described. These constructions are possible only for special choices of  $n$ , as shown in Table 1.

The best constructions for cubic graphs have diameter  $1.413 \log_2 n$  (see [7]). It is shown in [8] that a cubic graph obtained by adding a random perfect matching to a cycle has a diameter of order  $O(\log n)$ . In the same paper, the authors proved the following result:

**Theorem 1.1.** Suppose  $T$  is a complete binary tree on  $2^k - 1$  vertices. If we add two random matchings of size  $2^{k-2}$  to the leaves of  $T$ , then the resulting graph has diameter  $D$  satisfying

$$\log_2 n - 10 \leq D(G) \leq \log_2 n + \log_2 \log_2 n + 10$$

with probability approaching 1 as  $n$  approaches infinity.

In [9] it is shown that almost all  $k$ -regular graphs are hamiltonian for any  $k \geq 3$ , by an analysis of the distribution of 1-factors in random regular graphs.

In this paper we propose an algorithm for constructing hamiltonian graphs with  $n$  vertices, maximum degree  $\Delta$  and diameter bounded by:

$$D \leq 2 \cdot \lfloor \log_{\Delta-1} n \rfloor.$$

Our main contribution is that we assure diameter  $O(\log_{\Delta-1} n)$  for every  $n$ , not just for special values, while hamiltonicity and small diameter are achieved by using significantly fewer edges.

The paper is organized as follows. In Section 2 we describe the linear algorithm and prove that it produces hamiltonian graphs of degree at most  $\Delta$  and diameter at most  $2 \log_{\Delta-1} n$ . In Section 3 we deal with the case  $\Delta = 3$  and improve the upper bound for the number of edges in the resulting graph to  $\lfloor \frac{11n+6}{8} \rfloor$ . The same approach may be applied for  $\Delta > 3$  and this way we get a hamiltonian graph with the average degree asymptotically equal to  $4 - \frac{2}{\Delta-1} - \frac{2(\Delta-2)^2}{(\Delta-1)^3}$ . In the end, we show that the algorithm may be modified in such a way that it constructs a planar hamiltonian graph with degree at most  $\Delta$  and diameter at most  $2 \lfloor \log_2 n \rfloor$  and also point out some experimental results for diameter when  $\Delta = 3$ .

## 2. The algorithm

A complete binary tree is a tree with  $n$  levels, where for each level  $d \leq n - 1$ , the number of the existing nodes at level  $d$  equals  $2^d$ . This means that all possible nodes exist at these levels. An additional requirement for a complete binary tree is that for the  $n$ th level, while not all nodes have to exist, the nodes that do exist must fill the level from left to right (for more details see [10]). A complete binary tree is one of the most important architectures for interconnection networks [11].

A generalization of binary trees are  $\Delta$ -ary trees. Namely, every vertex has at most  $\Delta - 1$  children, and all vertices that are not in the last level have exactly  $\Delta - 1$  children. All vertices in the last level must occupy the leftmost spots consecutively.

We propose the following algorithm for constructing hamiltonian graphs of order  $n$ , maximum degree  $\Delta$  and diameter  $\leq 2 \log_{\Delta-1} n$ . First we construct a complete  $\Delta$ -ary tree with  $n$  vertices, and for every vertex we store its degree, parent, and all children from left to right. A labeled  $\Delta$ -ary tree contains labels 1 through  $n$  with root being 1, branches leading to nodes labeled 2, 3, ...,  $\Delta$ , branches from these leading to  $\Delta + 1$ ,  $\Delta + 2$ , ...,  $(\Delta - 1)(\Delta - 1) + \Delta$ , and so on. We also maintain a queue of all leaves.

We will traverse vertices in order to form a hamiltonian cycle. The starting vertex is the root of the tree. First we try to go up the tree through the parent of the current vertex – if the parent is already visited, we choose one of its children. In the case where all neighbors of the current vertex are marked, we pick an arbitrary unmarked leaf and add an edge connecting these two vertices.

**Algorithm 1:** Constructing a hamiltonian supergraph of a complete  $\Delta$ -ary tree

---

**Input:**  $\Delta$ -ary rooted tree with  $n$  nodes  
**Output:** Hamiltonian cycle in an array *cycle*,  $m$  is the total number of edges

```

1  $v = \text{root};$ 
2  $k = 1;$ 
3  $m = n - 1;$ 
4 while  $k \leq n$  do
5    $\text{cycle}[k] = v;$ 
6    $\text{mark}[v] = \text{true};$ 
7   if  $\text{parent}[v]$  is not marked then
8      $v = \text{parent}[v];$ 
9   else if there exists a child  $u$  from the set of neighbors which is not marked then
10     $v = u;$ 
11  else
12    Let  $u$  be a random unmarked leaf;
13    Add edge  $(v, u);$ 
14     $m = m + 1;$ 
15     $v = u;$ 
16  end
17   $k = k + 1;$ 
18 end
19 Add edge  $(v, \text{root});$ 
20  $m = m + 1;$ 

```

---

**Theorem 2.1.** A graph constructed by the above algorithm is hamiltonian.

**Proof.** The basic idea of the algorithm is to traverse the hamiltonian path by adding edges when they are needed. In the end, we join the last visited vertex to the root of the binary tree. There can be at most one vertex other than the root with a degree greater than 1 and less than  $\Delta$  in the tree – let this vertex have the label  $s$ . All edges that we add during the execution of the algorithm connect either two leaves or a leaf and  $s$ . By symmetry, we can eliminate the vertex  $s$  by traversing from the root to  $s$  in the first few steps of the algorithm. Therefore, this algorithm does not increase the maximum degree  $\Delta$  in the graph.

We cannot visit any vertex twice, so we have to prove that all vertices are marked. Assume that the vertex  $v$  is not visited and the algorithm has finished. If any vertex in its subtree is visited, then we would have already visited  $v$ , because we first try to go upwards. Therefore, all vertices in its subtree are unvisited. There is at least one leaf in the subtree, and we have to visit this leaf during the execution of the algorithm. This is a contradiction, so the constructed graph is hamiltonian.  $\square$

The diameter of this graph is less than or equal to twice the depth of the tree.

$$D(G) \leq 2 \cdot \lfloor \log_{\Delta-1} n \rfloor.$$

**Lemma 2.1.** The number of leaves  $L(n)$  in a complete  $\Delta$ -ary tree is

$$L(n) = \begin{cases} n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor, & \text{if } n \equiv 1 \pmod{(\Delta-1)} \\ n - 1 - \left\lfloor \frac{n}{\Delta-1} \right\rfloor, & \text{if } n \not\equiv 1 \pmod{(\Delta-1)}. \end{cases}$$

**Proof.** We know that  $L(1) = 1$  and  $L(k) = k - 1$  for  $k = 2, 3, \dots, \Delta - 1$ . Whenever we properly add  $\Delta - 1$  vertices to this tree, we always get exactly  $\Delta - 2$  new leaves. This proves the recurrent relation for the number of leaves

$$L(n) = L(n - \Delta + 1) + \Delta - 2.$$

By mathematical induction, one can easily prove the explicit formula for  $L(n)$ .  $\square$

**Theorem 2.2.** The number of edges in the constructed graph is less than  $(2 - \frac{1}{\Delta-1})n + \frac{\Delta-3}{2}$ .

**Proof.** There are  $\frac{n\Delta}{2}$  edges in a  $\Delta$ -regular graph. The internal nodes in a  $\Delta$ -ary tree are nodes with degree greater than one. After running the algorithm, every leaf will have degree at most three and every internal node will have degree at most  $\Delta$ . This gives an upper bound for the number of edges in the hamiltonian graph constructed by the algorithm:

$$|E| < \frac{n\Delta}{2} - \frac{\Delta-3}{2} \left( n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor - 1 \right) \leq \frac{3n}{2} + \frac{\Delta-3}{2} \left( \frac{n}{\Delta-1} + 1 \right) = \left( 2 - \frac{1}{\Delta-1} \right) n + \frac{\Delta-3}{2}. \quad \square$$

This bound is less than  $\frac{5n}{3}$  for  $\Delta = 4$ , and less than  $2n$  if  $\Delta > 4$  and the depth of the tree is greater than two. Based on this fact, we give the following.

**Proposition 2.1.** *Time and memory complexity of the proposed algorithm is linear  $O(n)$ .*  $\square$

### 3. A finer bound on the number of extra edges

Cubic graphs are of special interest in token ring topologies and because of their importance, in this section we first improve the previous estimations for the number of edges added in the construction of a hamiltonian path for the case  $\Delta = 3$  and then observe that the same approach may be applied to  $\Delta \geq 4$  as well.

First we examine graphs with  $n = 2^{k+1} - 1$  vertices, where  $k \geq 1$ . A corresponding binary tree has complete last level and there are  $2^k$  leaves in the tree. Let  $f(k)$  be the number of additional edges added to this tree, when we start the execution of the algorithm in an arbitrary leaf and end it in another leaf. One can easily verify that  $f(1) = 0$  and  $f(2) = 2$ . In addition, we define  $f(0) = 0$ . After traversing upwards to the root and then downwards to an arbitrary leaf, the graph induced by the unvisited vertices are disjoint complete binary trees of heights  $0, 1, \dots, k-2$ . Therefore, we may write the recurrent formula:

$$f(k) = 2(k-1) + 2 \cdot \sum_{i=0}^{k-2} f(i).$$

By strong induction we will prove that:

$$f(k) = \begin{cases} \frac{2}{3}(2^k - 1), & \text{if } k \text{ is even} \\ \frac{2}{3}(2^k - 2), & \text{if } k \text{ is odd.} \end{cases}$$

Assume that the formula holds for all numbers  $m < 2k$ , and now we will prove it for  $2k$  and  $2k+1$ .

$$f(2k) = 2(2k-1) + 2 \cdot \sum_{i=0}^{k-1} f(2i) + 2 \cdot \sum_{i=1}^{k-1} f(2i-1) = 4k-2 + \frac{4}{3} \cdot \sum_{i=0}^{2k-2} 2^i - \frac{4}{3} \cdot (3k-2) = \frac{2}{3} \cdot (2^{2k} - 1).$$

$$f(2k+1) = 2(2k+1-1) + 2 \cdot \sum_{i=0}^{k-1} f(2i) + 2 \cdot \sum_{i=1}^k f(2i-1) = 4k + \frac{4}{3} \cdot \sum_{i=0}^{2k-1} 2^i - \frac{4}{3} \cdot 3k = \frac{2}{3} \cdot (2^{2k+1} - 2).$$

To estimate the actual number of additional edges in the proposed algorithm we have to start from the root and add an additional edge connecting the last visited vertex and the root of the binary tree. After visiting the first leaf we have that the unvisited vertices form disjoint complete binary trees of heights  $0, 1, 2, \dots, k-2$  and  $k-1$ . So, the total number of edges to be added equals:

$$f(0) + f(1) + \dots + f(k-1) + k = \frac{f(k+1)}{2} = \left\lfloor \frac{n}{3} \right\rfloor.$$

Therefore, we proved the following result.

**Theorem 3.1.** *The number of additional edges in the case of a complete binary tree with  $2^k - 1$  vertices does not depend on the choice of random leaves in the algorithm and this number equals  $\frac{f(k+1)}{2}$ .*  $\square$

Now assume that  $n$  is not of the form  $2^{k+1} - 1$ . The main result in this section is the following theorem.

**Theorem 3.2.** *For every integer  $n \in \mathbb{N}$ , there exists a hamiltonian graph with diameter at most  $2 \lfloor \log_2 n \rfloor$ , maximum degree 3 and at most  $\lfloor \frac{11n}{8} + \frac{3}{4} \rfloor$  edges.*

**Proof.** The number of leaves in a corresponding binary tree with  $n$  vertices is  $\lceil \frac{n}{2} \rceil$ . Some leaves are on the last level, and some are on the level before last. Consider the consecutive leaves in the last level starting from the left and group them into groups of size four. This way we get subtrees of height two and there can be at most three unpaired leaves. We will do the same thing for the leaves in the previous level, but starting from the right.

Therefore, the number of subtrees of height two is less than or equal to  $\frac{n}{8}$  and greater than  $\frac{n-2.3}{8}$ . According to the two cases in Fig. 2, there can be two or three leaves in each of these subtrees that will get and keep degree two until the algorithm finishes. Thus, the number of edges we do not have to add equals half the number of vertices of degree two, and it is between  $\frac{n}{8} - \frac{3}{4}$  and  $\frac{3n}{16}$ . The total number of edges in the constructed graph after running the algorithm will be between  $\lfloor \frac{21n}{16} \rfloor$  and  $\lfloor \frac{11n}{8} + \frac{3}{4} \rfloor$ .  $\square$

**Remark 3.1.** After the hamiltonian cycle is constructed, we may insert additional edges into the graph to make it cubic, provided  $n$  is even.

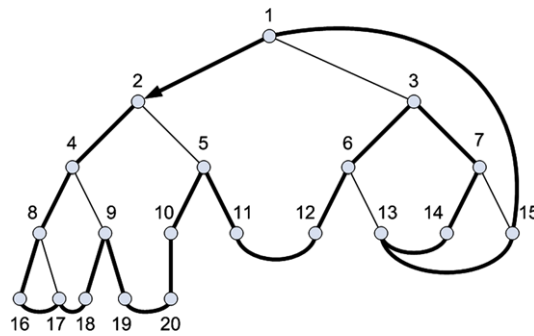


Fig. 1. Example of algorithm execution for a graph with  $n = 20$ ,  $\Delta = 3$  and vertex 1 as the root.

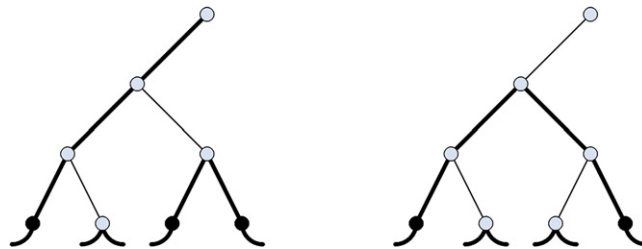


Fig. 2. Two cases for subtrees of height two.

For the case  $\Delta > 3$ , one can construct a hamiltonian graph with at most

$$\left(2 - \frac{1}{\Delta - 1} - \frac{(\Delta - 2)^2}{(\Delta - 1)^3}\right)n + \frac{\Delta - 3}{2} + 2(\Delta - 2)$$

edges. The same approach may be applied to complete binary subtrees of greater heights to obtain a slightly finer bound for the total number of edges.

#### 4. Concluding remarks

The proposed algorithm may be modified to construct a planar hamiltonian graph.

**Theorem 4.1.** *By appropriately choosing descendant and unvisited leaves in the algorithm, one can assure that the constructed graph is planar.*

**Proof.** In order to prove the theorem, we will construct a hamiltonian path that starts at the leftmost leaf and ends in the nearest leaf (neighboring leaf or leaf that is at distance three from it). For small values of  $n$ , this can be easily verified. In the general case we first go upwards to the root and then to the rightmost leaf. Now, the  $\Delta$ -ary tree is partitioned into smaller trees, which will be traversed by induction from left to right. These binary trees do not have vertices in common, and we can independently add necessary edges which do not intersect the existing edges. We reduce our problem to the previous case by going to the leftmost leaf. Now we have disjoint trees and we traverse them starting from the leaves. Finally, we add the last edge without intersection problems (as in Fig. 1).  $\square$

For the case  $\Delta = 3$ , in our implementation we always choose a leaf that is farthest from the current leaf. This heuristic is done by the breadth first search. We use only three arrays of length  $n$ , so memory requirements are linear in  $n$ . Time complexity is  $O(n^2)$ , because the number of edges is  $m \leq \frac{3n}{2} = O(n)$ . The diameters of the examples of cubic graphs constructed by this algorithm are shown in Fig. 3, where the  $x$ -axis carries  $n/2$  and  $n$  is the number of nodes. It is obvious from this figure that the constant 2 from our bound is not the best possible.

Instead of choosing the leaves at random, we can do it more sophisticatedly and further reduce the diameter of the graph. One possible way is to add a matching which connects only the leaves on the last level in the root's left and right subtrees. This way we can still apply the algorithm, but if we have to choose a random leaf to continue—we first check whether the paired leaf in the matching is marked. This way we decrease the diameter by a constant, which is at least one. We leave for future study to see whether this approach may be used to construct 1-edge hamiltonian graphs.

#### Acknowledgements

The authors are grateful to the reviewers for their valuable comments and suggestions.

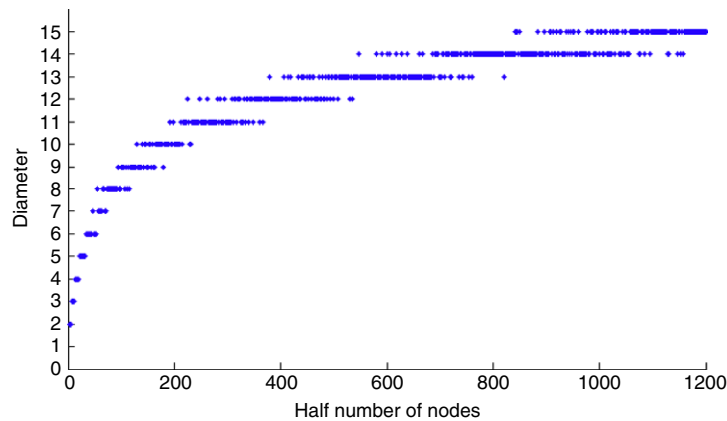


Fig. 3. The size of diameter for  $n = 4$  to  $n = 2400$  for  $\Delta = 3$ .

## References

- [1] J.C. Bermond, F. Comellas, D.F. Hsu, Distributed loop computer networks: A survey, *Journal of Parallel and Distributed Computing* 24 (1995) 2–10.
- [2] F. Harary, J.P. Hayes, Node fault tolerance in graphs, *Networks* 27 (1996) 19–23.
- [3] J.-J. Wang, C.N. Hung, L.H. Hsu, Optimal 1-hamiltonian graphs, *Information Processing Letters* 65 (1998) 157–161.
- [4] J.-J. Wang, T.-Y. Sung, L.-H. Hsu, M.-Y. Lin, A new family of optimal 1-hamiltonian graphs with small diameter, in: *Proceedings of the 4th Annual International Conference on Computing and Combinatorics*, 1998, pp. 269–278.
- [5] C.-N. Hung, L.-H. Hsu, T.-Y. Sung, Christmas tree: A versatile 1-fault-tolerant design for token rings, *Information Processing Letters* 72 (1999) 55–63.
- [6] S.-S. Kao, L.-H. Hsu, Brother trees: A family of optimal  $1_p$ -hamiltonian and 1-edge hamiltonian graphs, *Information Processing Letters* 86 (2003) 263–269.
- [7] M. Capalbo, An explicit construction of lower diameter cubic graphs, *SIAM Journal on Discrete Mathematics* 16 (2003) 630–634.
- [8] B. Bollobás, F.R.K. Chung, The diameter of a cycle plus a random matching, *SIAM Journal on Discrete Mathematics* 1 (1988) 328–333.
- [9] R.W. Robinson, N.C. Wormald, Almost all regular graphs are hamiltonian, *Random Structures and Algorithm* 5 (2) (1992) 363–374.
- [10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, second edn, MIT Press, 2001.
- [11] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.
- [12] M. Miler, J. Širáň, Moore graphs and beyond: A survey of the degree/diameter problem, *The Electronic Journal of Combinatorics* (2005) #DS14.